PROPOSED DISCUSSION

The following includes a proposed discussion for the Formal Interview set for 11:00 a.m. (EST) on June 29, 2005. The Proposed Discussion is drafted in light of the current Office Action dated June 3, 2005.

I.   Rejection Under 35 U.S.C. § 112, First Paragraph

Claims 17-21 are rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. Specifically, it is asserted that claim 17 contains a new matter, "an assignment of an address of a first variable," which is not supported by the specification. In response, applicants assert that the element, "an assignment of an address of a first variable," is supported by the specification. The specification recites as follows:

> "Figure 4A shows a graph following the next sequence of processing. The graph 400 includes elements similar to those in Figure 2A. The graph 400 also includes a ghost of a node 414 whose content includes *the address of the variable y.* The purpose of the node 414 is to aid the discussion to follow." (emphasis added).

> "Figure 4B shows a graph following the next sequence of processing. For illustrative purposes only, suppose an assignment statement defines "x=&y" in the program. The symbol "&" is understood to mean the inclusion of a unary operator in a programming language *to obtain an address of a variable.* Thus, for illustrative purposes only, the expression "&y" can be thought to be equivalent to a pointer to the variable y since this pointer *would contain an address of the variable y."* (emphasis added).
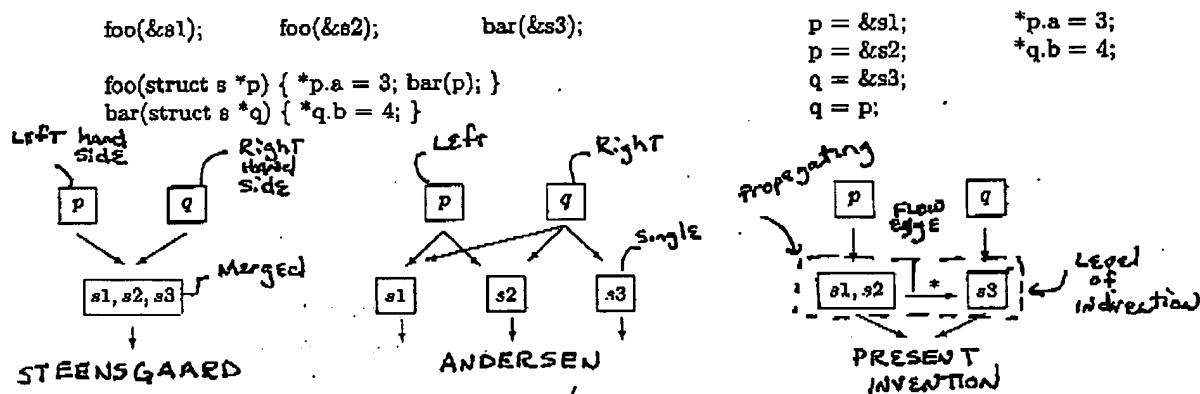
In light of the above, applicants believe that the element "an assignment of an address of a first variable" is properly supported. Accordingly, in that claims 17-21 are not further rejected, applicants believe that claims 17-21 are in condition for allowance.

App. No. 09/489,878
Proposed Discussion For Interview of June 29, 2005

II.   Rejection Under 35 U.S.C. § 103(a)

Claims 1-16, 22-36 and 52-56 are rejected under 35 U.S.C. 103(a) as being unpatentable

over "Points-to Analysis in Almost Linear Time" by Bjarne Steensgaard (hereinafter

"Steensgaard") in view of "Program Analysis and Specialization for the C Programming

Language" by Lars Ole Andersen (hereinafter Andersen). Applicants respectfully disagree with

the rejection. In hopes of clarifying several points, applicants propose the following Figure and

accompanying explaination.

A.   Explanation of Steensgaard, Andersen and the Present Invention

foo(&s1);   foo(&s2);   bar(&s3);   p = &s1;   *p.a = 3;
                                                                   p = &s2;   *q.b = 4;
foo(struct s *p) { *p.a = 3; bar(p); }   q = &s3;
bar(struct s *q) { *q.b = 4; }   q = p;

The above Steensgaard figure shows the points-to information computed by

Steensgaard's algorithm for the program in the Figure. The points-to graph shown contains

nodes representing equivalence classes of symbols, and edges representing pointer relationships.

Every node contains a single pointer edge. Stensgaard's algorithm processes assignments

bidirectionally: The left hand side and right hand side memory locations in an assignment are

Page 12 of 17

constrained to hold the same contents. For the example program, the effect of the assignment from p to q is to force both p and q to point to the equivalence class containing $s1$, $s2$, and $s3$, even though p cannot point to $s3$ in any execution of the program. Steensgaard's algorithm uses type equality rules to merge equivalence classes of symbols at assignments, leading to nodes with single outdegree in the points-to graph. The use of Steensgaard's analysis only allows for an approximation. Such an approximation may be insufficient in many cases.

The above Andersen figure shows the points-to information computed by Andersen's algorithm. In this case, every node is associated with a single symbol, and contains a set of pointer values. Assignments are processed directionally: The contents of the right hand side location are copied to the left hand side. This method produces expensive points-to sets as a result. Andersen's algorithm achieves directionality in assignments by using subtyping rules. In order to accommodate directionality, it is necessary to allow unlimited fanout, or outdegree, in the points-to graph. This leads to trees with fanout at all directions. The algorithm is expensive because of the work required to track the subtyping relations induced at all levels of the points-to graph.

The above figure for the present invention shows a method for enhancing pointer analysis. The pointer analysis includes propagating a label of the first one of the two locations to a label of the second one of the two locations such that the label of the first one of the two locations is a subset of the second one of the two locations (shown by the "*" flow edge). The two locations are selected to be one level of indirection away for a level associated with the assignment. Thus, the pointer target location of p contains $s1$ and $s2$, whereas the pointer target location of q contains $s1$ and $s2$, as well as $s3$. In this manner, the present invention allows

App. No. 09/489,878
Proposed Discussion For Interview of June 29, 2005

unlimited outdegree in critical areas, while restricting outdegree in other non-critical areas. The

present invention is, therefore, inexpensive and accurate in use.

Aspects of these elements are specifically recited in the claims. Claim 1 recites the

following elements that are not taught or suggested in the above-cited references:

> "processing an assignment between two variables in a program, wherein
> processing an assignment includes *forming a relationship between two locations
> that are related to the two variables, wherein the two locations are selected to be
> one level of indirection away from a level associated with the assignment,
> wherein each location includes a label* and a content, and wherein a content of a
> first one of the two locations is selectively unified with a content of a second one
> of the two locations" (emphasis added).

> "*propagating a label of the first one of the two locations to a label of the second
> one of the two locations such that the label of the first one of the two locations
> is a subset of the second one of the two locations*" (emphasis added).

Claim 5 recites the following elements that are not taught or suggested in the above-cited

references:

> "processing an assignment between two variables in a program, wherein
> processing an assignment includes *forming a relationship between two locations
> that are related to the two variables, wherein the two locations are selected to be
> one level of indirection away from a level associated with the assignment,*
> wherein each location includes a label and a content, and wherein a content of a
> first one of the two locations is selectively unified with a content of a second one
> of the two locations" (emphasis added).

> "*propagating a label of the first one of the two locations to a label of the second
> one of the two locations such that the label of the first one of the two locations
> is a subset of the second one of the two locations*" (emphasis added).

Claim 6 recites the following elements that are not taught or suggested in the above-cited

references:

> "*defining a relationship between two locations upon an assignment in the
> program, wherein the two locations are selected to be one level of indirection
> away from a level associated with the assignment, wherein a label of a first one
> of the two locations is associated with a label of a second one of the two*

Page 14 of 17

locations, and wherein contents of the two locations are selectively unified" (emphasis added).

Claim 12 recites the following elements that are not taught or suggested in the above-cited references:

"*forming a relationship between two locations upon an assignment of a first variable and a second variable in the program, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment*, wherein the relationship defines that a label of *a first of the two locations is a subset of a label of a second of the two locations*, and wherein contents of the two locations are selectively unified" (emphasis added).

Claim 22 recites the following elements that are not taught or suggested in the above-cited references:

"*forming a relationship between two locations upon an assignment of a first variable and a dereference of a second variable in the program, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment, wherein the relationship defines that a label of a first of the two locations is a subset of a label of a second of the two locations*, and wherein contents of the two locations are selectively unified" (emphasis added).

Claim 27 recites the following elements that are not taught or suggested in the above-cited references:

"*forming a relationship between two locations upon an assignment of a dereference of a first variable and a second variable in the program, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment, wherein the relationship defines that a label of a first of the two locations is a subset of a label of a second of the two locations*, and wherein contents of the two locations are selectively unified" (emphasis added).

Claim 32 recites the following elements that are not taught or suggested in the above-cited references:

"*defining a relationship between two locations upon an assignment in the program, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment, wherein a label of a first of*

the two locations is defined as a subset of a label of a second of the two locations, and wherein contents of the two locations are selectively unified" (emphasis added).

Claim 52 recites the following elements that are not taught or suggested in the above-

cited references:

"processing a plurality of assignment statements in a program to derive a plurality of sets of information, wherein the plurality of sets of information *is distributed among a plurality of levels of indirection*" (emphasis added).

"establishing a plurality of flow relationships corresponding to each of the plurality of assignment statements, wherein each of the flow relationships is selected to be established one level of indirection away from each of the assignment statements" (emphasis added).

Claim 55 recites the following elements that are not taught or suggested in the above-
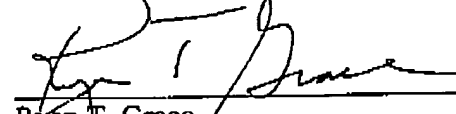
cited references:

"an analyzer to analyze the tree to produce an object file, wherein the object file contains at least one relationship between two variables in an assignment statement in the program, *wherein the relationship defines that a set of symbols relating to one of the two variables is a subset of a set of symbols relating to the other of the two variables, and wherein another relationship is selectively formed one level of indirection away from a level associated with the assignment statement between the set of symbols related to one of the two variables and the set of symbols relating to the other of the two variables*" (emphasis added).

Accordingly, applicants propose that the elements of independent claims 1, 5, 6, 12, 17, 22, 27,

32, 52, and 55 are not taught or otherwise suggested by the cited references. In sofar as the

remaining claims depend from independent claims 1, 5, 6, 12, 17, 22, 27, 32, 52, and 55, they are

thought to be allowable for at least those same reasons. Applicants' attorney respectfully

requests discussion of the above issues during the June 29, 2005 interview.

App. No. 09/489,878
Proposed Discussion For Interview of June 29, 2005

Respectfully,

MERCHANT & GOULD P.C.

Ryan T. Grace
Registration No. 52,956
Direct Dial: 206.342.6258

MERCHANT & GOULD P.C.

27488
PATENT TRADEMARK OFFICE